State Space Gaussian Process Prediction

Zitao Liu ZTLIU@CS.PITT.EDU
Lei Wu LEIWU@CS.PITT.EDU
Milos Hauskrecht MILOS@CS.PITT.EDU
Department of Computer Science, University of Pittsburgh, PA 15213 USA

Abstract

Learning accurate models of complex clinical time-series data is critical for understanding the disease and its dynamics. Modeling of clinical time-series is particularly challenging because: observations are made at irregular time intervals and may be missing for long periods of time. In this work, we propose a new model of clinical time series data that is optimized to handle irregularly sampled and missing observations. Our framework combines two models: the linear state-space and the Gaussian Processes (GP) models, into a novel dynamical model, named State Space Gaussian Process (SSGP). The model is learned using the expectation-maximization algorithm that iterates between inferences in the dynamic model and learning of the parameters of the underlying SSGP dynamic model. Experiments on real-world clinical time-series data show that the model outperforms alternative time-series prediction models.

1. Introduction

Accurate modeling of clinical time-series data is extremely important for disease prediction and patient management. The modeling of clinical time series comes with a number of challenges (Reis & Mandl, 2003; Combi et al., 2010; Batal et al., 2011). In this work we focus on two modeling issues. First, the time-series for an individual patient may vary in length and may span a few or many days depending on the length of patient's hospitalization. Second, the time-series observations are obtained at different times which means the time elapsed between the two consecutive observations may vary. Our objective is to devise dynamic models and algorithms that are flexible enough to work under these assumptions.

A typical and widely applied time-series prediction model is the linear state-space model (Kalman, 1963). The advantage of the model is its simplicity and existing algorithms to learn the model from observational data. However, the linear state-space models are somewhat limited and may not work well on clinical time-series data with possible nonlinearities and with missing and irregularly sampled observations.

The Gaussian process (GP) model is a non-linear non-parametric model defining the distribution over functions $f(x)$ (Rasmussen & Williams, 2006). The model is robust for predicting a function value f for any datapoint x and this even in the presence of a substantial noise. Assuming x represents the time one would hope the GP would let us model time-series of observations at irregular time intervals. However, the application of the GP to the clinical time domain is not straightforward. First, it is not clear how one should define the mean function of the GP that is flexible enough for the clinical time series prediction. Second, the mean function in general depends on the time which raises the question of how to allign the different clinical time-series data (corresponding to different patients). One way to alleviate the problem is to assume that the mean function defining the process is constant in time. However, this assumption is too simple and it would greatly limit the GP application in temporal domains in which the data may change in time.

In this work, we address the above problems by proposing and testing a new model - the State Space Gaussian Process (SSGP) for time-series analysis and prediction. The model combines the linear state-space model and the Gaussian process model into a single framework. Briefly, we define the mean function of the Gaussian process as a linear combination of basis functions that is restricted to the time window of a limited size. The Markov Chain, modeled by the linear state-

Appearing in the *ICML 2012 Workshop on Clinical Data Analysis*, Edinburgh, Scotland, UK, 2012.

space model, controls the evolution (or changes) of basis function weights of the mean function in between two consecutive windows. This combination is non-trivial for two reasons. First, the new system permits us to define more complex mean functions anchored at the time window origin. Second, the optimization of the combined system is quite different than the optimization of each sub-system.

To learn the parameters of the SSGP model, we propose a new algorithm that is an extension of the well-known Expectation-Maximization algorithm (Dempster et al., 1977) used to learn parameters of probabilistic models with hidden variables. We test the model and the algorithm on the problem of time-series prediction for six common blood tests from the complete blood count (CBC) panel. Our results show that our model leads to a more accurate predictive performance than alternative time-series models. In addition, our model is more robust than the alternatives when the number of patients and observations used to train the models is small.

The main contribution of this work is a more flexible time-series prediction model that can handle irregularly-sampled observation sequences. Instead of applying a one-step look-ahead forecasting model to make stepwise predictions, our multi-step predictor lets us predict future values for a certain time horizon' more robustly. In addition, the model is less sensitive to the error accumulation problem that often affects time prediction models (Cheng et al., 2006).

Our paper is structured as follows. First, in Section 2, we cover the basics of linear state space model and Gaussian processes. In Section 3, we formulate the problem we want to solve and describe our state-space Gaussian process model. Section 4 and Section 5 explain the inference and learning details in SSGP. Section 6 describes the general procedure for applying the SSGP to predict future time-series values. Section 7 focuses on regression experiments on both the synthetic and the real-clinical data, and comparison to alternative modeling approaches. Finally, Section 8 summarizes the work and outlines possible future extensions.

2. Background

Linear State Space Model

The time-invariant discrete linear state space model, also known as the Linear Dynamical System (LDS), is a classic and widely used real-valued time series model (Vladimir Pavlovic & MacCormick, 2000; Carvalho & Lopes, 2007; Sang Min Oh & Dellaert, 2008). A LDS on variables $\mathbf{x}_{1:T}, \mathbf{y}_{1:T}$ is defined in terms of the following two equations:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + \mathbf{w}_t \; ; \; \mathbf{y}_t = C\mathbf{x}_t + \mathbf{v}_t$$

where $t \in \{1, \ldots, T\}$ is the discrete time index; x_1 is the initial state distribution with mean μ and covariance V, $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{x}_1|\mu, V)$, $\mathbf{x}_t$ are the hidden states generated by the transition matrix A with independent zero mean noise $\mathbf{w}_t, \mathbf{w}_t \sim \mathcal{N}(\mathbf{w}_t|0, Q)$; and $\mathbf{y}_t$ are the observations generated by the emission matrix C with independent variate noise $\mathbf{v}_t, \mathbf{v}_t \sim \mathcal{N}(\mathbf{v}_t|0, R)$. LDS is characterized by a state transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ where $p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t|A\mathbf{x}_{t-1}, Q)$, and a state to observation probability $p(\mathbf{y}_t|\mathbf{x}_t)$ where $p(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t|C\mathbf{x}_t, R)$. The complete parameter set is $\Theta = \{A, C, Q, R, \mu, V\}$. The EM algorithm is widely used for estimating the parameters of LDS (Ghahramani & Hinton, 1996).

Gaussian Process

The Gaussian process ($\mathcal{GP}$) model is a popular non-parametric nonlinear Bayesian models for machine learning. $\mathcal{GP}$ is an extension of multivariate Gaussians to infinite-sized collections of real-valued variables. We can think of this extension as the distributions over random functions (Rasmussen & Williams, 2006). A $\mathcal{GP}$ is specified by its mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and its covariance function $K(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$ where $f(\mathbf{x})$ is the real process. Since $\mathcal{GP}$ can be viewed as a Gaussian distribution over functions, it can be used to estimate the values of some function f at arbitrary position x_*. This application is referred to as *Gaussian Process Regression* (Rasmussen & Williams, 2006; Quionero-Candela & Rasmussen, 2005). The basic GP regression equations are

$$\bar{f}_* = K(x_*, \mathbf{x}) \left[K(\mathbf{x}, \mathbf{x}) + \sigma^2 I\right]^{-1} \mathbf{y}$$

$$Cov(f_*) = K(x_*, x_*) - K(x_*, \mathbf{x}) \left[K(\mathbf{x}, \mathbf{x}) + \sigma^2 I\right]^{-1} K(x_*, \mathbf{x})$$

where I is the identity matrix, $\mathbf{x}$ is the input vector and $\mathbf{y}$ is the output or target, $\bar{f}_*$ is the posterior function mean and $Cov(f_*)$ is the posterior covariance. With the right choice of the covariance function, the associated prediction uncertainty increases in regions away from observations while it shrinks when it is close to observed data. This property makes the $\mathcal{GP}$ model very appealing for time series prediction.

Although the GP model has a great potential to capture the fluctuation and variation in the time series domain, it is unclear how one should define its mean function. Another problem is the allignment of the different clinical time series and the decision on how to define the time origin. To avoid the problem one

may set the mean function to a constant value, but this choice significantly limits the ability of the model to capture the variations observed in clinical data.

3. State Space Gaussian Process

3.1. Problem Formulation

We aim to learn a time series prediction/regression function $g : \mathbf{X} \rightarrow \mathbf{Y}$ where $\mathbf{X}$ is composed of two parts $\mathbf{V}$ and $\mathbf{T}$: $\mathbf{X} = \{\mathbf{V}, \mathbf{T}\}$. $\mathbf{V}$ is the set of initial observations, and each observation is a *time-value* pair. $\mathbf{T}$ is an *arbitrary* $1 \times n$ time index vector $\mathbf{T} = \{t_1, t_2, \ldots, t_n\}$, where n is the number of time steps we want to predict, and $\mathbf{Y}$ is the $1 \times n$ predicted value vector of the future observations at $\mathbf{T}$, i.e. $\mathbf{Y} = \{Y_{t_1}, Y_{t_2}, \ldots, Y_{t_n}\}$. Typically, we learn such a function by using values at regularly sampled discrete time points, which means $t_{i+1} - t_i = C$, where $1 \leq i \leq N-1$, N is the size of training data set and C is a constant reflecting the data sampling interval.

Here, we assume our observation are sampled irregularly from some process which is more general and common in real-life settings. For example, in clinical domain, if the observations are some lab test values for a patient during his or her hospital period, those test values are often recorded irregularly due to different patients' health conditions.

3.2. The Model

We start the description of our model by first describing its Gaussian process component. More specifically, we consider the Gaussian process $g(\mathbf{x})$ whose mean function is a combination of a few fixed basis functions with coefficients, $\boldsymbol{\beta}$.

$$g(\mathbf{x}) = f(\mathbf{x}) + \mathbf{h}(\mathbf{x})^T\boldsymbol{\beta} \text{ where } f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$$

Here $f(\mathbf{x})$ is a zero mean $\mathcal{GP}$, $\mathbf{h}(\mathbf{x})$ are a set of fixed basis functions, i.e. $\mathbf{h}(x) = (1, x, x^2, \ldots)$. and $\boldsymbol{\beta}$ is a Gaussian prior, $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{b}, I)$. Following (O'Hagan & Kingman, 1978) we can obtain another $\mathcal{GP}$

$$g(\mathbf{x}) \sim \mathcal{GP}(\mathbf{h}(\mathbf{x})^\mathbf{T}\mathbf{b}, k(\mathbf{x}, \mathbf{x}') + \mathbf{h}(\mathbf{x})^T\mathbf{h}(\mathbf{x}')).$$

The above Gaussian process definition (with single $\boldsymbol{\beta}$) may not be flexible enough for the entire time series. In addition, the mean function of the process depends on time (and the time origin) which makes it hard to allign time series for multiple patients. To achieve more flexibility we assume the above Gaussian process represents the time-series only in the time window of a limited duration, and that the dynamics of the entire time-series is captured by a linear state space model representing the unknown transition of $\boldsymbol{\beta}$ for two consecutive time windows. This allows us to represent the entire time series variations in a more flexible manner.

More specifically, we chop the entire irregular time series data into m windows $\mathbf{w} = \{w_1, w_2, \ldots, w_m\}$ for a fixed window size W. For each window w_i, we use $Y_{i,:}$ to represent all the observations that fall into window w_i and $Y_{i,j}$ is the jth observation in w_i. Instead of using a single GP to capture the variation in the entire time series, we divide the responsibility into different windows, and each window w_i is associated with one GP. We use $\boldsymbol{\beta}_i$ to denote the Gaussian prior coefficients in GP_i's mean function, $\boldsymbol{\beta}_i \sim \mathcal{N}(\mathbf{b}_i, I)$, which is unknown. And we set a first order hidden Markov chain $\mathbf{Z} \equiv \{Z_i\}$ to model the transition between $\boldsymbol{\beta} \equiv \{\boldsymbol{\beta}_i\}$, which forms a linear state space system: $Z_{t+1} = AZ_t + \mathbf{w}_t$ and $\boldsymbol{\beta}_t = CZ_t + \mathbf{v}_t$ where $\mathbf{w}_t$ and $\mathbf{v}_t$ are zero-mean normally distributed random variables with covariance matrices Q and R respectively. Since the entire time series data is generated from certain stochastic process, we assume different windows' GPs share the same covariance function, which is parametrized by Θ.

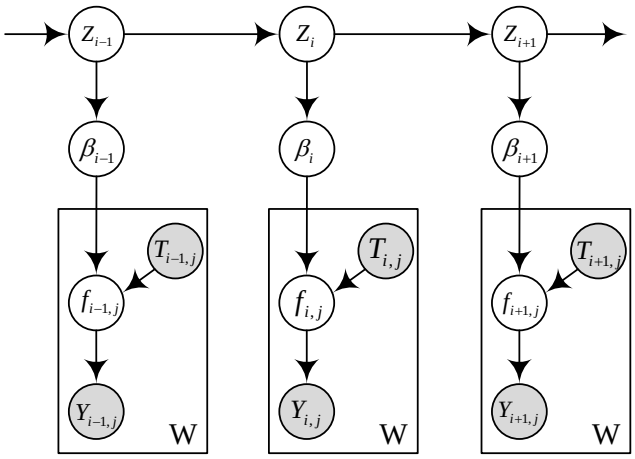

Figure 1. State space Gaussian process model for multi-step prediction. The shaded nodes $Y_{i,j}$ denote the irregular observations and shaded nodes $T_{i,j}$ denote the time information of each observation. Each plate corresponds to a chopped window, which is associated with a GP. W is the number of observations in each window. $f_{i,j}$ is Gaussian field.

The initial state distribution is also learned during the training. Just as with the linear state space model, the prior on the initial state is a Gaussian with mean π_1 and covariance V_1. The hyper parameters of mean function for the $\{GP_i\}$ are denoted by $\{\mathbf{b}_i\}$. The entire parameter space can be summarized as $\Omega := \{\Theta, \{\mathbf{b}_i\}, A, C, R, Q, \pi_1, V_1\}$ and the graphical representation of our state space Gaussian process model is shown in Figure 1. When the horizontal ar-

rows are removed, breaking the time dynamics, the graphical model reduces to a set of independent Gaussian process regression models. With time dynamics, the coefficients of mean function at slice i has smoothly evolved from those at slice i-1.

3.3. Choice of the Covariance Function

Mean Reverting Mean-reverting process is also called Ornstein-Uhlenbeck process, which is stationary, Gaussian, and Markovian, and it satisfies these three conditions, up to allowing linear transformations of the space and time variables (Gillespie, 1996; Doob, 1942). Over time, the process tends to drift towards its long-term mean. Mean reverting is an important property in clinical time series prediction. Since different patients have different values and variations, but in the long run, the values should approach to the long-term mean. In the covariance function, we reflect the mean reverting phenomenon by $K_1 = \sigma_1 \exp(\theta_1 |\mathbf{x} - \mathbf{x}'|)$.

Periodicity In clinical domain, the time series always explicitly or implicitly reflect the periodic information. And the periodic form can capture the fluctuation within the short period well. Another major benefit is that the periodic function can keep the variation of different values within the reasonable range. For these reasons, we use the periodic covariance function $K_2 = \sigma_2 \exp(\theta_2 \sin^2\left[\frac{\omega}{2\pi}(\mathbf{x} - \mathbf{x}')\right])$.

In this work, we choose $K = K_1 + K_2$ as our $\mathcal{GP}$'s covariance function. However, the choice for covariance function is quite flexible.

4. Inference(E-step)

Since both the Markov hidden chain $\{Z_i\}$ and the mean coefficient $\{\boldsymbol{\beta}_i\}$ are unobserved, we cannot learn $\{GP_i\}$ directly; instead, we apply the EM algorithm (Dempster et al., 1977) to learn linear hidden transition of GPs' mean coefficients and its covariance hyper parameter together.

The E-step infers a posterior distribution of latent states $\mathbf{Z}, \boldsymbol{\beta}$ given the observation sequences $\mathbf{Y} = \{Y_{i,:}\}$, $p(\mathbf{Z}, \boldsymbol{\beta} | \mathbf{Y}, \Omega)$. In the following, we omit the explicit conditioning on Ω for notational brevity. Due to the conditional independence encoded in SSGP, the joint distribution of the data is given by:

$$
\begin{aligned}
p(D) &= p(\mathbf{Z}, \boldsymbol{\beta}, \mathbf{Y}) \\
&= p(Z_1) \prod_{i=2}^{m} p(Z_i | Z_{i-1}) \prod_{i=1}^{m} (\boldsymbol{\beta}_i | Z_i) \prod_{i=1}^{m} \prod_{j=1}^{W} p(Y_{i,j} | \boldsymbol{\beta}_i)
\end{aligned}
$$

This E-step requires computing the expected log likelihood $\mathcal{Q} = \mathbb{E}_{\boldsymbol{\beta},\mathbf{Z}}[\log p(\boldsymbol{\beta}, \mathbf{Z}, \mathbf{Y} | \Omega)]$, which is depends on $\mathbb{E}[Z_i|\mathbf{Y}]$, $\mathbb{E}[Z_i Z_i^{'}|\mathbf{Y}]$ and $\mathbb{E}[Z_i Z_{i-1}^{'}|\mathbf{Y}]$. Let define $\hat{Z}_{i|T} \equiv \mathbb{E}[Z_i|\mathbf{Y}]$, $M_{i|T} \equiv \mathbb{E}[Z_i Z_i^{'}|\mathbf{Y}]$, $M_{i,i-1|T} \equiv \mathbb{E}[Z_i Z_{i-1}^{'}|\mathbf{Y}]$, $P_{i|T} = \mathbb{VAR}[Z_i|\mathbf{Y}]$ and $P_{i,i-1|T} = \mathbb{VAR}[Z_i Z_{i-1}^{'}|\mathbf{Y}]$. T is the length of time series. Note that the hidden state estimate $\hat{Z}_{i|T}$ depends on both past and future observations.

To compute $\hat{Z}_{i|T}$ and $M_{i|T}$, we follow (Shumway & Stoffer, 1982) performing a backward algorithm to compute these hidden state estimations given on all(previous, current, and future) observations. See details in Algorithm 1.

5. Learning(M-step)

In the following, we derive the M-step for gradient based optimization of the parameters Ω. In the M-step, we try to find Ω that maximize the likelihood lower bound $\mathcal{Q} = \mathbb{E}_{\boldsymbol{\beta},\mathbf{Z}}[\log p(\boldsymbol{\beta}, \mathbf{Z}, \mathbf{Y} | \Omega)]$. In the following, we omit the explicit conditioning on Ω for notational brevity. The factorization properties of SSGP yield the decomposition $\mathcal{Q}$ into

$$
\begin{aligned}
&\mathcal{Q} = \mathbb{E}_{\mathbf{X},\mathbf{Z}}[\log p(\boldsymbol{\beta}, \mathbf{Z}, \mathbf{Y})] = \mathbb{E}_{\boldsymbol{\beta},\mathbf{Z}}[\log p(Z_1)] \\
&+\mathbb{E}_{\boldsymbol{\beta},\mathbf{Z}}\Big[\sum_{i=2}^{m} \log p(Z_i | Z_{i-1})\Big] + \mathbb{E}_{\boldsymbol{\beta},\mathbf{Z}}\Big[\sum_{i=1}^{m} \log p(\boldsymbol{\beta}_i | Z_i)\Big] \\
&+\mathbb{E}_{\boldsymbol{\beta},\mathbf{Z}}\Big[\sum_{i=1}^{m} \sum_{j=1}^{W} \log p(Y_{i,j} | \boldsymbol{\beta}_i)\Big] \qquad (1)
\end{aligned}
$$

As we can see from eq. 1, the shares parameters Θ of the covariance function for all $\{GP_i\}$ only appear in the last term of $\mathcal{Q}$, which is $\mathbb{E}_{\boldsymbol{\beta},\mathbf{Z}}\Big[\sum_{i=1}^{m} \sum_{j=1}^{W} \log p(Y_{i,j} | \boldsymbol{\beta}_i)\Big]$. So that we can easily get the derivative(see eq. 2) and use any gradient based optimizer to estimate them.

For each of the rest of parameters $\{\{\mathbf{b}_i\}, A, C, R, Q, \pi_1, V_1\}$, we re-estimate them by taking the corresponding partial derivative of the expected log likelihood, setting to zero, and solving. These result in Algorithm 2.

Summary of the Algorithm

The manifestation of parameter estimation in the SSGP is summarized by Algorithm 3. Let define $\hat{\mathbf{Z}}_i \equiv \{\hat{Z}_{i|T}\}_1^T$, $\mathbf{M}_i \equiv \{M_{i|T}\}_1^T$ and $\mathbf{M}_{i,i-1} \equiv \{M_{i,i-1|T}\}_1^T$. The function *SSGPsmoother* implements the E-step and the *maximize* routine implements the M-step.

$$
\begin{aligned}
\frac{\partial \log p(\mathbf{Y}|\Theta)}{\partial \Theta} &= -\frac{1}{2}\mathrm{Tr}\left[K^{-1}\frac{\partial K}{\partial \Theta}\right] \\
&+ \frac{1}{2}\mathbf{Y}^T K^{-1} \frac{\partial K}{\partial \Theta} K^{-1}\mathbf{Y} \qquad (2)
\end{aligned}
$$

6. Prediction

For the multi-step prediction, let us suppose our training data is $\mathbf{Y} = \{Y_1, Y_2, \ldots, Y_N\}$, which are all the time series sequences' time-value pairs, and our objective is, given few observations $\mathbf{V}$ and arbitrary time index $\mathbf{T} = \{t_1, t_2, \ldots, t_n\}$, to predict $\{Y_{t_1}, \ldots, Y_{t_n}\}$.

Algorithm 1 EM: E-step

Backward algorithm for SSGP:
// Compute $\hat{Z}_{i|T}$, $M_{i|T}$ and $M_{i,i-1|T}$
// By definition, $M_{i|T} = P_{i|T} + \hat{Z}_{i|T}\hat{Z}_{i|T}^{'}$
// By definition, $M_{i,i-1|T} = P_{i,i-1|T} + \hat{Z}_{i|T}\hat{Z}_{i-1|T}^{'}$
// Initialization: $P_{T,T-1|T} = (I - K_T C)AP_{T-1|T-1}$
$J_{i-1} = P_{i-1|i-1}A^{'}(P_{i|i-1})^{-1}$
$\hat{Z}_{i-1|T} = \hat{Z}_{i-1|i-1} + J_{i-1}(\hat{Z}_{i|T} - A\hat{Z}_{i-1|i-1})$
$P_{i-1|T} = P_{i-1|i-1} + J_{i-1}(P_{i|T} - P_{i|i-1})J_{i-1}^{'}$
$P_{i-1,i-2|T} = P_{i-1|i-1}J_{i-2}^{'} + J_{i-1}(P_{i,i-1|T} - AP_{i-1|i-1})J_{i-2}^{'}$
// where $P_{i-1|i-1}, P_{i|i-1}, \hat{Z}_{i|i-1}, \hat{Z}_{i|i}$ and K_i are computed by *Kalman_Filter*. See Algorithm 5 in Appendix A.1.

Algorithm 2 EM: M-step

//Define H_i matrix collects the $\mathbf{h}(\mathbf{x})$ vectors for all the observations $Y_{i,:}$ in window w_i.
for $i = 1$ **to** m **do**
  $E_i = (K_{Y_{i,:}} + H_i^{'}H_i)^{-1}$
  $\mathbf{b}_i = (R^{-1} + H_i E_i H_i^{'})^{-1}(R^{-1}C\hat{Z}_{i|T} + H_i E_i Y_{i,:})$
end for
$\pi_1 = \hat{Z}_{1|T}$
$V_1 = M_{1|T} - \hat{Z}_{1|T}\hat{Z}_{1|T}^{'}$
$A = (\sum_{i=2}^{m} M_{i,i-1|T})(\sum_{i=2}^{m} M_{i-1|T})^{-1}$
$Q = \frac{1}{m-1}(\sum_{i=2}^{m} M_{i|T} - A\sum_{i=2}^{m} M_{i-1,i|T})$
$R = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{b}_i\mathbf{b}_i^{'} - C\hat{Z}_{i|T}\mathbf{b}_i^{'})$
$C = (\sum_{i=2}^{m} \mathbf{b}_i\hat{Z}_{i|T}^{'})(\sum_{i=1}^{m} M_{i|T})^{-1}$

Algorithm 3 Parameter Estimation in SSGP

Get Θ by any gradient optimizer based on eq. 2.
init $\Omega\backslash\Theta$
repeat
  E-step: Section 4, Algorithm 1
  $\hat{\mathbf{Z}}_i$, $\mathbf{M}_i$ and $\mathbf{M}_{i,i-1} \leftarrow$ *SSGPsmoother* $(\mathbf{Y}, \Omega\backslash\Theta)$
  M-step: Section 5, Algorithm 2
  $\Omega\backslash\Theta \leftarrow$ *maximize* $\mathcal{Q}(\Omega, \hat{\mathbf{Z}}_i, \mathbf{M}_i, \mathbf{M}_{i,i-1})$ wrt $\Omega\backslash\Theta$
until Convergence
return $\Omega = \{\Theta, \{\mathbf{b}_i\}, A, C, R, Q, \pi_1, V_1\}$

We first adopt *Kalman Filter* to get the estimation of the hidden state Z, given the initial observations $\mathbf{V}$ and the learned parameters $\{A, C, R, Q, \pi_1, V_1\}$. Then we split $\{Y_{t_1}, \ldots, Y_{t_n}\}$ into windows, for each window w_i, we use the linear equation $\boldsymbol{\beta}_i = CZ_i + v_i$ to obtain the estimation of the mean parameter and use GP_i to make the multi-step prediction during this window w_i. Last, we repeat the above steps and do prediction window by window. The prediction can be summarized in Algorithm 4.

Algorithm 4 Prediction with SSGP

// Data preprocessing
Split the predicted data $\{Y_{t_1}, \ldots, Y_{t_n}\}$ into l windows $\{w_1, \ldots, w_l\}$.
// Estimate hidden state by initial observations
// See *Kalman_Filter* in Appendix A.1
$Z = Kalman\_Filter(\mathbf{V}, A, C, R, Q, \pi_1, V_1)$
// Do prediction window by window
for $i = 1$ **to** l **do**
  $\boldsymbol{\beta}_i = CZ + \mathcal{N}(0, R)$
  $\mathbf{b}_i = \mathbb{E}[\boldsymbol{\beta}_i]$
  $\{\hat{Y}_{i,1}, \hat{Y}_{i,2}, \ldots, \hat{Y}_{i,w}\} = \mathcal{GP}(\mathbf{h}(\mathbf{x})^{\mathbf{T}}\mathbf{b}_i, k(\mathbf{x}, \mathbf{x}') + \mathbf{h}(\mathbf{x})^{T}\mathbf{h}(\mathbf{x}'))$
  // Update the hidden state
  $Z = AZ + \mathcal{N}(0, Q)$
end for

7. Experiments and Results

In this experiment we test our method on data from electronic health records of approximately 4,500 post-surgical cardiac patients stored in PCP database (Hauskrecht et al., 2010; Valko & Hauskrecht, 2010). To test the performance of our multi-step prediction model, we select 1800 patients who took the *Complete Blood Count*(CBC) test[1] and use a 5-fold cross validation on datasets of different sizes (from 400 to 1800 with a step of 200). In this experiment, we picked these *six* tests in CBC as our dataset, see Table 1.

Table 1. Six lab test in CBC.

Name	Explanation
HCT	Red blood cells percentage in blood
MCHC	Average hemoglobin concentration
MCV	Average size of patient's red blood cell
MPV	Average size of patient's platelets
RBC	Actual count of red blood cells
PLT	Platelets counts in a given blood

These time series data are noisy, their signal fluctu-

[1]CBC test is used as a broad screening test to check for such disorders as anemia, infection, and many other diseases

ate in time, and observations are obtained with varied time-interval period. Figure 2 illustrates such time series. The X-axis is the time index aligned by hour and the Y-axis is the values/observations for each test.

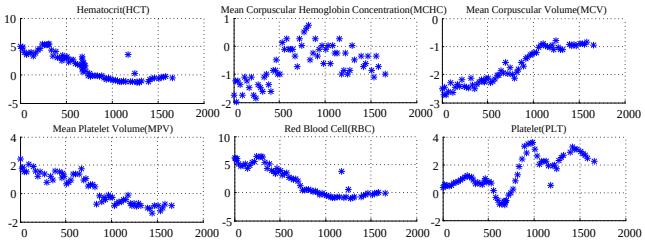

Figure 2. Six time series test samples from one patient's Complete Blood Count(CBC) test.

We evaluated our model on this real clinical data sets using multi-step-ahead prediction. We compared SSGP predictions to three other methods: (1) Gaussian Process regression(GP) with covariance function $K(\mathbf{x}, \mathbf{x}') = \sigma_1 \exp(\theta_1 |\mathbf{x} - \mathbf{x}'|) + \sigma_2 \exp(\theta_2 \sin^2\left[\frac{\omega}{2\pi}(\mathbf{x} - \mathbf{x}')\right])$; (2) Linear Dynamical Model(LDS) trained with the entire time series observations; (3) Linear Dynamical Model trained with the k recent time series observations(KNN-LDS).

For *LDS* and *KNN-LDS*, we pre-define the time interval and use discretization to handle the sample-irregularity problem. For the empty time intervals, we apply the Gaussian process interpolation to fill the missing values (Gibbs & MacKay, 1997; Li & Orchard, 2001).For *KNN-LDS*, we set $k = 15$ in this setting.

We evaluate and compare the performances of the different methods by calculating the Root Mean Square Error(RMSE) on the test data. Root Mean Square Error is employed as the performance measurements to compare our proposed approach with the above state-of-the-art prediction/regression methods. More specifically, the metrics RMSE is defined as follows:

$$RMSE = \left[n^{-1} \sum_{i=1}^{n} |y_i - \hat{y}_i|^2 \right]^{1/2}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value and n is the number of data points.

The results of RMSE on 6 CBC test samples are summarized in Figure 3.

Discussion

The results of our experiments show that in general, our state space Gaussian process(SSGP) model significantly outperform all other methods in terms of prediction errors on all six CBC lab tests. One of the advantages of our method is that its prediction error is small even when it is trained on a small number of patients and observations. Specifically, from Figure 3, we find the following results:

First, when comparing *GP, SSGP* to *LDS, KNN-LDS*, we can see that the continuous methods (GP, SSGP) apparently outperform the discretized methods (LDS,KNN-LDS). The reasons are 1) the values from patients' tests are always around a normal range plus some variation. The combination of the mean reverting function and the periodic function captures this phenomenon: the mean reverting function forces the predicted values within a normal range and the periodic function allows the fluctuation and variation flexibility. Clearly, LDS and KNN-LDS cannot capture these variations by their linear equations. 2) LDS and KNN-LDS solve the multi-step prediction problem by constructing a single model from past observations and by predicting the future values iteratively. Since they use predictions from the past, they are very susceptible to the *error accumulation*: errors generated in the history are propagated into future predictions (Cheng et al., 2006). The*GP, SSGP* make the multi-step prediction directly and hence suffer less from this problem. For the MCV lab test result, we can see the *GP* performs worse than *LDS* and *KNN-LDS*. We explain this by the fact that the MCV time series vary very little, and the linear model can fit the data well. On the other hand, the *GP* is more complex and may overfit the data and outliers present in the data.

Second, compared with other methods, *SSGP* does not require a large number of training examples and it can perform well even with small training data. However, the error rates of other methods are decreased by a large amount due to the decrease in the number of examples. In the clinical domain, dataset availability is a big issue. The data is very expensive to obtain. Stable performance on small-size training data is very important in practice.

Third, comparing *GP* and *SSGP*, we can see, the *SSGP* is much better than the *GP* approach. It reduces 30%-60% of the errors compared with the GP approach. It shows that a single constant mean is not enough in the complex time series settings. The evolution of mean variables in the consecutive windows is modelled by a linear dynamical system, which expresses a stronger descriptive ability. During the prediction phase, its predicted mean is used by the subsequent GP to make more accurate predictions.

Fourth, compared with the performance of *LDS* and *KNN-LDS*, we can find that KNN-LDS, in most cases, obtains a lower error than the generic LDS which shows that the entire time series data are not always

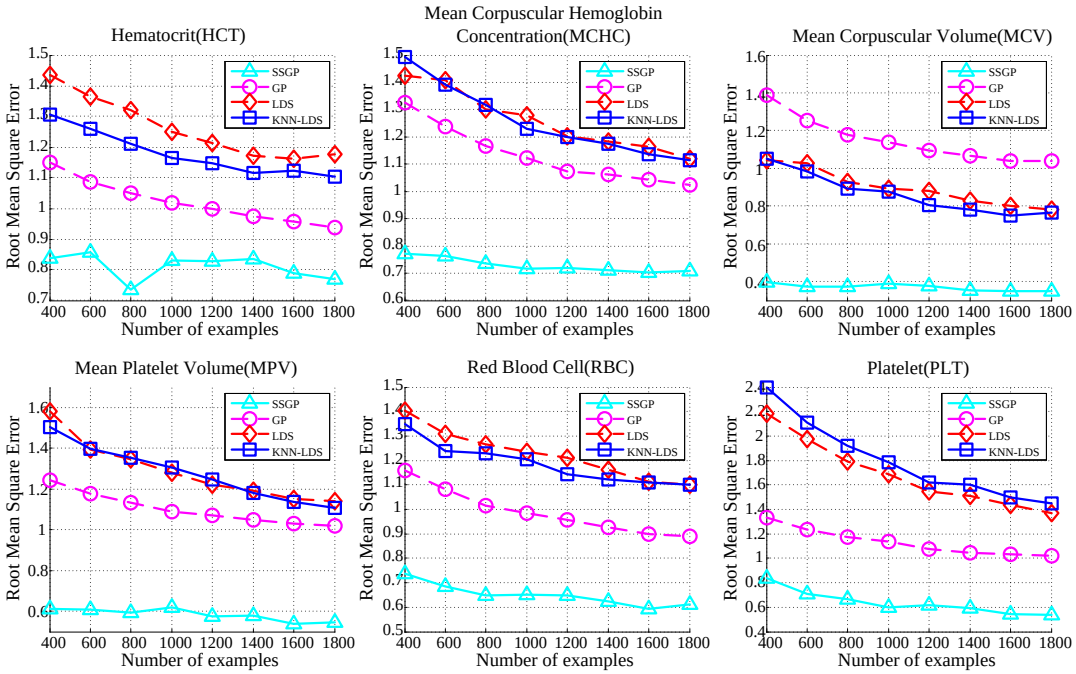

Figure 3. Root Mean Square Error(RMSE) on CBC test samples.

useful. In the clinical domain, the more recent values we observe, the more important they are for predicting the future values.

8. Conclusion

In this paper, we have presented a state space Gaussian process system for the multi-step prediction. Comparing with the traditional linear state space systems and modern Gaussian process regression, special features of this novel system are its robustness to irregular sampling and small training sequence data, which are very important in clinical monitoring and alerting systems. Another advantage of the system is its ability to make good long-term multi-step predictions. Experimental results on real world clinical data from electronic health records systems demonstrated that the novel prediction model achieves errors that statistically significant lower than errors for other state of the art approaches used in clinical time sequence data prediction. In the future, we plan to study and consider dependences among multiple time series, and extend the framework to multivariate time series modeling. Other extensions we plan to study include switching-state (Vladimir Pavlovic & MacCormick, 2000) and controlled (Hauskrecht & Fraser, 1998; Kveton & Hauskrecht, 2006) dynamical systems.

Acknowledgments

This work was supported by grants 1R01LM010019-01A1 and 1R01GM088224-01 from the NIH. Its content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

Batal, Iyad, Valizadegan, Hamed, Cooper, Gregory F, and Hauskrecht, Milos. A pattern mining approach for classifying multivariate temporal data. In *IEEE International Conference on Bioinformatics and Biomedicine*, pp. 358–365. IEEE, 2011.

Carvalho, C.M. and Lopes, H.F. Simulation-based sequential analysis of markov switching stochastic volatility models. *Computational Statistics & Data Analysis*, 51(9):4526–4542, 2007.

Cheng, H., Tan, P.N., Gao, J., and Scripps, J. Multistep-ahead time series prediction. *Advances in Knowledge Discovery and Data Mining*, pp. 765–774, 2006.

Combi, Carlo, Keravnou-Papailiou, Elpida, and Shahar, Yuval. *Temporal information systems in medicine*. Springer Publishing Company, Incorporated, 2010.

Dempster, Arthur P, Laird, Nan M, and Rubin, Donald B. Maximum likelihood from incomplete data

via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

Doob, J. L. The brownian movement and stochastic equations. In *Annals of Mathematics*, volume 43, pp. 351–369. April 1942.

Ghahramani, Zoubin and Hinton, Geoffrey E. Parameter esimation for linear dynamical systems. Technical report, Department of Computer Science, University of Toronto, 1996.

Gibbs, Mark and MacKay, David J.C. Efficient implementation of gaussian processes. Technical report, 1997.

Gillespie, Daniel T. Exact numerical simulation of the ornstein-uhlenbeck process and its integral. *Phys. Rev. E*, 54:2084–2091, Aug 1996.

Hauskrecht, M., Valko, M., Batal, I., Clermont, G., Visweswaran, S., and Cooper, G.F. Conditional outlier detection for clinical alerting. In *AMIA Annual Symposium Proceedings*, pp. 286 – 290, 2010.

Hauskrecht, Milos and Fraser, Hamish. Modeling treatment of ischemic heart disease with partially observable markov decision processes. In *Proceedings of the AMIA Symposium*, pp. 538 – 542. American Medical Informatics Association, 1998.

Kalman, R. E. A new approach to linear filtering and prediction problem. In *Transactions of the ASME-Journal of Basic Engineering*, number 82 in D, pp. 35–45. 1960.

Kalman, R. E. Mathematical description of linear dynamical systems. *SIAM Journal on Control*, 1:152–192, 1963.

Kveton, Branislav and Hauskrecht, Milos. Solving factored mdps with exponential-family transition models. In *16th International Conference on Automated Planning and Scheduling*, pp. 114–120, 2006.

Li, Xin and Orchard, Michael T. New edge-directed interpolation. In *IEEE Transactions on Image Processing*, volume 10, pp. 1521–1527. October 2001.

O'Hagan, A. and Kingman, J. F. C. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40:1–42, 1978.

Quionero-Candela, Joaquin and Rasmussen, Carl Edward. A unifying view of sparse approximate gaussian process regression. In *JMLR*. 2005.

Rasmussen, Carl Edward and Williams, Christopher K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Reis, Ben Y and Mandl, Kenneth D. Time series modeling for syndromic surveillance. *BMC Medical Informatics and Decision Making*, 3, 2003.

Sang Min Oh, James M. Rehg, Tucker Balch and Dellaert, Frank. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. In *IJCV*. 2008.

Shumway, R. H. and Stoffer, D. S. An approach to time series smoothing and forecasting using the em algorithm. *Journal of Time Series Analysis*, 3:253–264, 1982.

Valko, Michal and Hauskrecht, Milos. Feature importance analysis for patient management decisions. In *MEDINFO*, 2010.

Vladimir Pavlovic, James M. Rehg and MacCormick, John. Learning switching linear models of human motion. In *NIPS*, 2000.

Welch, Greg and Bishop, Gary. An introduction to the kalman filter. Technical Report TR 95-041, University of North Carolina at Chapel Hill, 2006.

Appendix

A.1 Kalman Filter Inference

Algorithm 5 Kalman_Filter (Kalman Filtering Inference by Kalman (Kalman, 1960) (Welch & Bishop, 2006))

// Input $A, C, R, Q, \pi_1, V_1, \{Y_t\}$ and $\{Z_t\}$ is the hidden state.
// $\hat{Z}_{t|t-1} = \mathbb{E}[Z_t|\{Y_i\}_1^{t-1}]$ is the *priori* estimation and $P_{t|t-1} = \mathbb{E}[(Z_t - \hat{Z}_{t|t-1})(Z_t - \hat{Z}_{t|t-1})^T]$ is the *priori* estimate error covariance.
// $\hat{Z}_{t-1|t-1} = \mathbb{E}[Z_{t-1}|\{Y_i\}_1^{t-1}]$ is the *posteriori* estimation and $P_{t-1|t-1} = \mathbb{E}[(Z_{t-1} - \hat{Z}_{t-1|t-1})(Z_{t-1} - \hat{Z}_{t-1|t-1})^T]$ is the *posteriori* estimate error covariance.
// Time Update:
$\hat{\mathbf{Z}}_{t|t-1} = A\hat{\mathbf{Z}}_{t-1|t-1}$
$P_{t|t-1} = AP_{t-1|t-1}A^T + Q$
// Measure Update:
$K_t = P_{t|t-1}C^T(CP_{t|t-1}C^T + R)^{-1}$
$\hat{\mathbf{Z}}_{t|t} = \hat{\mathbf{Z}}_{t|t-1} + K_t(Y_t - C\hat{\mathbf{Z}}_{t|t-1})$